**《数据库原理与技术》课程报告**

**航空管理系统优化报告**

**学 院：电子信息与电气工程学院**

**专 业：软件工程**

**小组成员：** 窦嘉伟 秦梁 沈佳威 付玉晗

1. **业务背景简述**

设计一个航空出行管理系统，涉及系统管理员、乘客、地勤三种角色每个角色可能有多个用户。

系统管理员可以导入航班、票价等信息；

地勤可以提供值机服务，包括但不限于检查乘客行李是否有违禁品（含有违禁品的在处理完成前不得登机）、记录行李重量、登机前的信息确认等；

乘客可以查询航班、订购机票（需要体现乘客的账户余额变更）、选座、查询个人飞行记录等。

航班到达时，系统管理员可以确认航班到达（更新航班状态等信息）；

航班延误或取消时，系统管理员可以发布对应通知，对延误的情况为乘客提供部分补偿，对取消的情况提供全额补偿。

1. **优化目标描述**

使用sql子查询优化、索引、分表等技术实现大量数据的查询性能优化。

1. **性能分析与测试过程以及相关测试结果**

性能分析：观察优化前调用的查询语句运行的时间。

优化前使用的测试数据源代码以及脚本见脚本附件，针对每个实体集插入大约30万条数据。

主要优化的查询函数有：

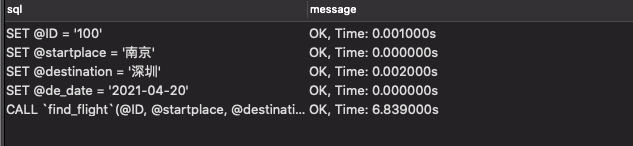
record\_weight（记录行李重量）

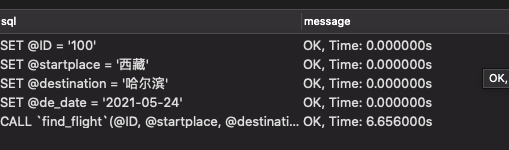
find\_flight（用户查票）

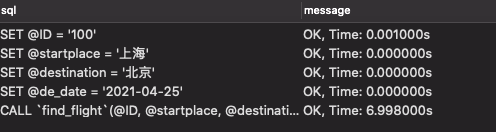
ADD\_flight（增加航班）

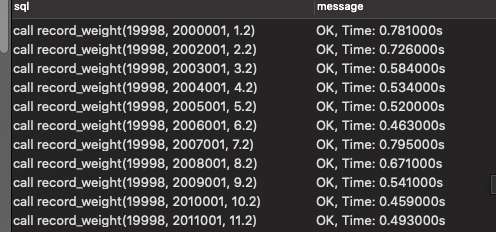
buy\_ticket（买票）等。

优化前测试过程以及结果见下图：

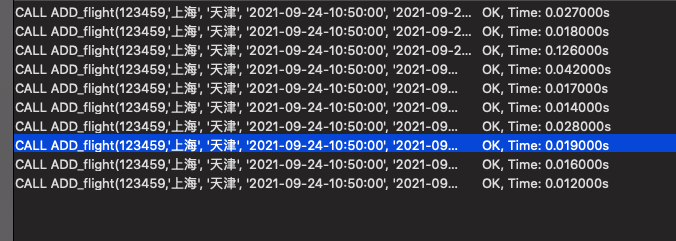












1. **优化方法描述及优化前后性能的对比**

优化方法：添加索引，更改了sql子查询，使用了数据库分页技术。

更改子查询和使用数据库分页技术的前后对比：见附件（函数优化.docx），例如：

ADD\_flight

SELECT count(\*) INTO identity

FROM (`User` NATURAL JOIN `User\_Role`)NATURAL JOIN `Role`

WHERE `User`.`ID`=ID and `Role\_id`= 2 ;

优化join后为

SELECT count(\*) INTO identity

        FROM (select ID

              from `User`

              where `ID` = ID) as A

              NATURAL JOIN

              (select ID,Role\_id

               from `User\_Role`

               where `ID`=ID and `Role\_id`= 2

              ) as B

              NATURAL JOIN

              (select Role\_id

               from `Role`

               where 'Role\_id' = 2

              ) as C

优化分页后为

DROP PROCEDURE IF EXISTS `ADD\_flight`;

delimiter ;;

CREATE PROCEDURE `ADD\_flight`(IN ID VARCHAR(40),IN start\_city VARCHAR(40),IN dest\_city VARCHAR(40) ,IN start\_date datetime(6),IN dest\_date datetime(6))

BEGIN

    DECLARE m VARCHAR(40);

    DECLARE identity INT ;

    DECLARE identity1 INT ;

    DECLARE identity2 INT ;

        SELECT count(\*) INTO identity1

        FROM (select ID

              from `User`

              where `ID` = ID and `ID` >10000 and `ID` < 15000) as A

              NATURAL JOIN

              (select ID,Role\_id

               from `User\_Role`

               where `ID`=ID and `Role\_id`= 2 and `ID` >10000 and `ID` < 15000

              ) as B

              NATURAL JOIN

              (select Role\_id

               from `Role`

               where 'Role\_id' = 2

              ) as C;

        if identity1 < 1

        THEN

        SELECT count(\*) INTO identity2

        FROM (select ID

              from `User`

              where `ID` = ID and `ID` >=15000 and `ID` < 20000) as A

              NATURAL JOIN

              (select ID,Role\_id

               from `User\_Role`

               where `ID`=ID and `Role\_id`= 2 and `ID` >= 15000 and `ID` < 20000

              ) as B

              NATURAL JOIN

              (select Role\_id

               from `Role`

               where 'Role\_id' = 2

              ) as C;

        end if;

        set identity = identity1 + identity2;

        IF identity>=1

        THEN

    SELECT max(cast(`Flight\_id` AS UNSIGNED)) INTO m FROM `Flight`;

    INSERT INTO `Flight` VALUES(m+1,start\_city,dest\_city,NULL,start\_date,dest\_date ,NULL,NULL);

    CALL insert\_record('Add\_Flight',ID);

    INSERT INTO `Flight\_State` VALUES(m+1,'6');

    CALL insert\_record('Add Flight\_State',ID);

    END IF;

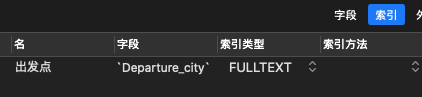
    END

;;

delimiter ;

添加索引：Flight表 添加flight\_id和departure\_city索引。ticket表添加ticket\_id和ID索引，例如：



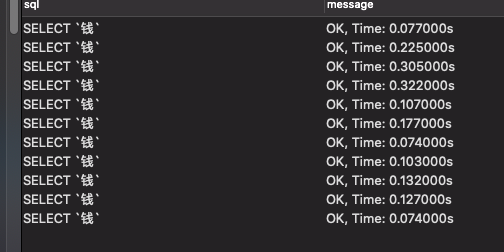


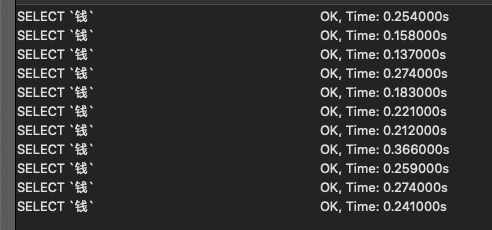
优化后函数性能如下图：

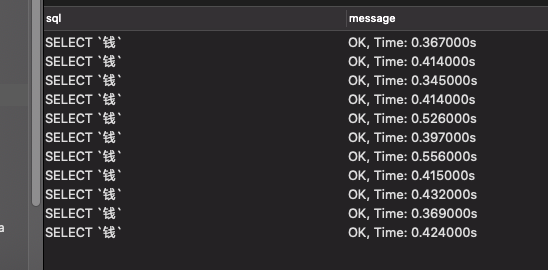


1. **加分项**

数据库引擎调研，memory引擎测试+innodb测试+MyISAM引擎。







数据库分页技术（见函数优化.docx附件）。

优化写入数据的性能，主要通过优化sql语句来实现。

